Acquisition of Named-Entity-Related Relations for Searching*

Tri-Thanh Nguyen and Akira Shimazu

Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
{t-thanh,shimazu}@jaist.ac.jp

Abstract. Named entities (NEs) are important in many Natural Language Processing (NLP) applications, and discovering NE-related relations in texts may be beneficial for these applications. This paper proposes a method to extract the ISA relation between a "*named entity*" and its *category*, and an IS-RELATED-TO relation between the *category* and its related object. Based on the pattern extraction algorithm "*Person Category Extraction*" (PCE), we extend it for solving our problem. Our experiments on Wall Street Journal (WSJ) corpus show promising results. We also demonstrate a possible application of these relations by utilizing them for semantic search.

Keywords: Named-entity-related relations extraction, information extraction, pattern extraction, algorithm, semantic search.

1. Introduction

Text documents often contain valuable relations among entities. For example, in the sentence taken from the WSJ corpus:

> There's a generally more positive attitude toward the economy, said Bette Raptapoulos, analyst for Prudential-Bache Securities Inc., ... (1)

there are relations: "Bette Raptapoulos" **is-a** analyst, and analyst **for** "Prudential-Bache Securities Inc."

Such relations may be beneficial in many NLP applications, such as for answering *Who* and *List* questions, e.g., "Who is *Bette Raptapoulos*?" or "Give me the list of *analyst for Prudential-Bache Securities Inc.*"

Relations in text documents can be extracted by pattern extraction as in (Brin 1998). (Brin 1998) presented Dual Iterative Pattern Relation Extraction (DIPRE), and used DIPRE to extract ⟨*author*, *title*⟩ tuples describing the relation: the author of the book *title* is *author*. Based on Brin's model, (Agichtein and Gravano 2000) presented the *Snowball* system to extract ⟨*organization*, *location*⟩ tuples indicating that the headquarters of *organization* is in *location*. (Nguyen and Shimazu 2007) developed the PCE system for extracting ⟨*person*, *category*⟩ tuples describing that the *person* is-a *category*.

This study proposes to automatically extract quadruples ⟨*ne*, *category*, *related-to*, *object*⟩ describing that the named entity *ne* ISA *category*, and the *category* IS-RELATED-TO *object*. We call such relations "*named-entity* is-a *category*" relation, and "*category* related-to *object*"

* This study was supported by Japan Advanced Institute of Science and Technology, the 21st Century COE Program: "Verifiable and Evolvable e-Society".

relation. The *related-to* and *object* of a quadruple may be null. We extend PCE algorithm to extract quadruples, and build a semantic search system to utilize extracted quadruples for answering some types of questions.
The remainder of this paper is organized as follows: Section 2 summarises some related work. Section 3 describes the original PCE algorithm and our extraction model. Section 4 gives a possible application of the extracted quadruples; Section 5 presents experiments and evaluation; Conclusions are given in the last section.

2. Related work

(Brin 1998) presented the DIPRE algorithm for extracting relations, and used DIPRE to extract (*author*, *title*) tuples having the relation: the author of the book *title* is *author*. Starting with a small number of (*author*, *title*) seed tuples, DIPRE finds the occurrences of tuples in order to generate new patterns. New patterns are, again, used to extract further (*author*, *title*) tuples. The DIPRE algorithm is graphically depicted in Figure 1.
Based on DIPRE, (Agichtein and Gravano 2000) introduced another method of generating new patterns, and developed the Snowball system for extracting (*organization*, *location*) tuples expressing the relation: the headquarters of *organization* is in *location*.
Current Named Entity Recognition (NER) systems often operate based on a predefined set of named entity classes, and assign a unique class to a discovered named entity (Chieu and Tou 2003). This is not natural, since the potential classes of named entity is large, and a named entity may belong to more than one class. For example, a person named entity may be both "*executive vice president*" and "*chief financial officer*" as expressed in the sentence: "Daniel Akerson, executive vice president and chief financial officer, said MCI's growth is being fueled by …" With the purpose of extending the number of named entity classes, (Nguyen and Shimazu 2007) proposed the "Person Category Extraction" (PCE) algorithm to automatically extract fine-grained categories of person Named Entities from text corpora. Based on DPIRE, (Nguyen and Shimazu 2007) introduced new types of patterns based on part-of-speech (POS) and chunk tags. One more proposal of their study which improved the perfomance of PCE a lot was the use of a validation function in the extraction procedure. Details of the PCE algorithm are provided in Section 3.1.

3. Extraction system

In this section, we describe the original PCE algorithm, and our extension for extracting ⟨*ne*, *category*, *related-to*, *object*⟩ quadruples.

4. PCE algorithm

The PCE algorithm is depicted in Figure 2, and the description is given in Figure 3. Starting with two seed *patterns*, PCE extracted ⟨*person*, *category*⟩ tuples. The extracted tuples were used to extract *occurrences* of ⟨*person*, *category*⟩ tuples in texts for generating new patterns. Again, new patterns were used to extract new tuples. The process terminated when no more patterns were produced. A pattern is defined as a 4-tuple:

(*order*, *person_slot*, *middle*, *category_pattern*),

where *order* (a Boolean value) indicates the occurrence order of *person* and

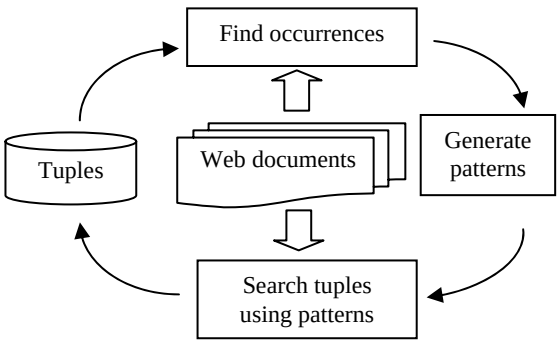


Figure 1: The DIPRE model.

category in a sentence; *person_slot* is a slot which will be replaced with a person named entity; *middle* is the string surrounded by *person* and *category*; *category_pattern* is defined as:

category_pattern:=*noun_phrase*$_1$ (and *noun_phrase*$_2$)?[1]

where *noun_phrase*$_i$ is a regular expression that matches a noun phrase with added POS tags.

In order to extract new ⟨*person*, *category*⟩ tuples, for every sentence *s*, from a pattern (whose *order* is true), and for each person NE *named_entity* in *s*, we construct a regular expression:

* *named_entity middle category_pattern* *

If *s* matches the above regular expression, the ⟨*person*, *category*⟩ tuple is extracted according to the algorithm in Figure 4, where *is_valid*(*category*) is a function that returns true if *category* is a sort of 'person', and false otherwise. The purpose of this function is to ignore unexpected matches, i.e., matches that give incorrect tuples. The *is_valid* function operates based on the fact that if a *person* is-a *category*, then the *category* must be a sort (or subtype) of person. Since a *category* is a sort of person is equivalent to the *category* is a *hyponym* of person (or person is a hypernym of the *category*), this constraint is checked by using WordNet (Fellbaum 1998) which contains hyponymic and hypernymic relations among concepts. When *order* is false, *named_entity* and *category_pattern* are switched. PCE can extract two tuples from a match, if there are two.

Occurrences: An occurrence of a ⟨*person*, *category*⟩ tuple is defined as a 4-tuple:

(*order*, *person*, *middle*, *category*),

where *middle* is a string surrounded by *person* and *category*. An occurrence of a (*person*, *category*) tuple is extracted if a sentence *s* matches the regular expression:

* *person middle category* *

or

* *category middle person* *

After extracting occurrences from the text corpus, they are used to generate new patterns. However, a *middle* of an occurrence is not necessarily reliable, (Nguyen and Shimazu) proposed a method to retain reliable ones based on two criteria: *repetition* and *diversity* as follows:

Repetition of a *middle* (repetition(*middle*)) is the number of times the *middle* appears between the *person* and *category* of (*person*, *category*) tuples of same *person*.

Diversity of a *middle* (diversity(*middle*)) is the number of times the *middle* appears between the *person* and *category* of (*person*, *category*) tuples of different *persons*.

A *middle* that has repetition(*middle*)>*threshold*$_R$ seems reliable and is kept. A pattern seems specific if it is generated based on tuples of a *person*, so only *middles* that have diversity(*middle*)>*threshold*$_D$ are kept to make the generated patterns general (Condition 1).

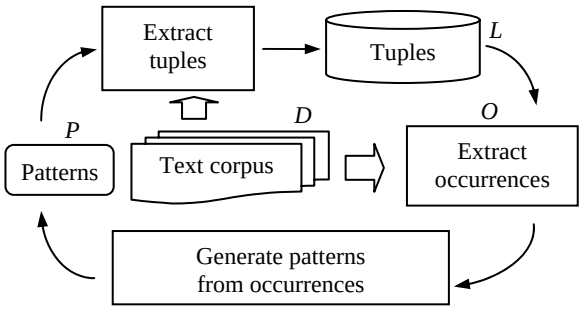


Figure 2: The PCE model.

[1] '?' stands for there is zero or one. Since PCE works on sentences which has been parsed by a shallow parser, each sentence contains POS and chunk tags. In this paper, we ignore POS and chunk tags in all regular expressions for readability.

If a *middle* contains a verb phrase, the verb phrase should express the relation *person* is-a *category* (Condition 2).

These two conditions are used in the pattern generation procedure as described in Figure 5. In the experiments, for the simplity, $threshold_R$ and $threshold_D$ are set to the same value which was called *threshold* for short.

Pattern types: Patterns whose *middles* are generated directly from the *middles* of occurrences are called *exact patterns*. Exact patterns are relately reliable, howerver, they have low coverage. In order to increase the coverage, (Nguyen and Shimazu 2007) introduced two more types of patterns, i.e., *sketch* and *extended sketch* patterns. An example of *middle* of an exact pattern with *order* true is:

"] ,/, [NP ABC/NNP] [NP 's/POS " (2)

The exact pattern with *middle* (2) can match the sentence:

[NP Harvey/NNP Dzodin/NNP] ,/, [NP ABC/NNP] [NP 's/POS vice/NN president/NN] … (3)

However, this pattern can not match a similar sentence that describes the "director" of another company, e.g., IBM, in the same syntax as (3):

Input: Text corpus D; Seed pattern set P_S
Output: The list L of quadruples
Preprocessing: Find NEs in every sentence by an NER;
    Remove sentences that contain no person NE.
    Add part-of-speech (POS) and chunk tags for every sentence by a shallow parser;
1. $P \leftarrow P_S$; $L \leftarrow \emptyset$;
2. Extract the list L' of quadruples from sentences that match a pattern in P; $L \leftarrow L + L'$;
    Let D' be list of sentences from which quadruples in L' were extracted, $D \leftarrow D - D'$;
    If D is empty then return;
3. Extract the list of occurrences O of quadruples in D;
4. Generate new patterns set P' from O; $P \leftarrow P'$;
    If P is empty then return; else go to Step 2;

Figure 3: PCE algorithm.

---

1. Generate $category_i$ by removing all POS tags in $noun\_phrase_i$.
2. If $is\_valid(category_i)$, then
    Generate person ne by removing all POS tags in $named\_entity$ to form a $\langle person, category_i \rangle$ tuple;
    Return $\langle person, category \rangle$ tuples;

Figure 4: ⟨*person*, *category*⟩ extraction from a match.

---

1. Group all occurrences in the list O by $order$ and $middle$; Let the resulting groups be $O_1, O_2, \ldots, O_N$;
2. For each group O_i, if the $middle$ satisfies the two conditions, then generate a new pattern:
    $(order, person\_slot, middle, category\_pattern)$

Figure 5: Pattern generation procedure.

[NP Alan/NNP Baratz/NNP] ,/, [NP IBM/NNP] [NP 's/POS director/NN] ... (4)

If *middle* (2) is modified so that its pattern can match (4), then expected relations in both (3) and (4) can be extracted. In order to do this, (2) is converted into a *template* that can match other sequences having similar structure. Concretely, nouns, adjectives, cardinals and articles in a *middle* are replaced with a variable $*word* that matches a word. Below is the template constructed from the *middle* (2):

"] ,/, [NP $*word*/NNP] [NP 's/POS " (5)

This template was called the *sketch* of a *middle*. A new pattern type whose the *middle* is replaced with a *sketch* was called *sketch pattern*. Details about the extended sketch patterns and other information can be seen in (Nguyen and Shimazu 2007).

4.1. Named entity category object extraction

The purpose of PCE is to extract ⟨*person*, *category*⟩ tuples, in which *category* can be used as the fine-grained type of person, so the set of NE types can be expanded by automatically extracting from texts. When we extract the tuple ⟨"Bette Raptapoulos", **'**analyst'⟩ from (1), we only have information: "Bette Raptapoulos" **is-a '**analyst'. If we can extract the relation: 'analyst' **for** "Prudential-Bache Securities Inc.", we will have complete information about "Bette Raptapoulos". Since PCE can be used to extract tuples of other NE types, such as *organization* and *location*, we propose to extend the PCE to extract ⟨*ne*, *category*, *related-to*, *object*⟩ quadruples describing the relations: *ne* is-a *category*, and *category related-to object* (or related-to relations for short).
From our observations, the related-to relations can be expressed in the following ways:

a) The *category* and *object* are linked by a preposition: "*category preposition object*", e.g., "analyst **for** Prudential-Bache Securities Inc."

b) The *category* and *object* are connected by a possessive apostrophe: "*object*'**s** *category*", e.g., "Semi-Tech'**s** chief executive officer". This can be interpreted as "*category* of *object*", e.g., "chief executive officer of Semi-Tech".

c) The *object* and *ne* are linked by a preposition: "*category ne preposition object*", e.g., "… said economist David Littmann of Manufacturers National Bank…", from which an expected quadruple is ⟨"David Littmann", 'economist', 'of', "Manufacturers National Bank"⟩.

d) The *object* is embedded in *category*, e.g., "IBM president". This can also be interpreted as "*category* of *object*", *e.g.*, "president of IBM".

e) The related-to relation is implicitly expressed, e.g., "Mr. Baird, who heads the Manhattan U.S. attorney's securities-fraud unit, denied the quote …", from which an expected quadruple is ⟨'Baird', 'header', 'of', "securities-fraud unit"⟩.

Since case e) does not have fixed expressions, we do not treat such cases. In case d), because *object* is already embedded in *category*, we do not need to extract the *object*. For cases a), b) and c), we build regular expressions to extract the *object*. We modify the procedure in Figure 4 to extract ⟨*ne*, *category*, *related-to*, *object*⟩ quadruples instead of ⟨*ne*, *category*⟩ tuples. Let *category_str* be the string containing the *category*, the regular expressions corresponding to each case are (we omit POS and chunk tags for readability):

a) * *category_str preposition noun_phrase* *

b) * *noun*_phrase**'s** *category_str* *

c) * *named_entity preposition noun_phrase* *

After extracting a valid *category* and an *ne* (Step 2 of Figure 4), if the current processing sentence matches one of the above regular expressions, the *object* is produced by removing POS tags in *noun_phrase*; *related-to* is the *preposition* after removing POS tags in cases a) and c); *related-to* is 'of' in case b), then, ⟨*ne*, *category*, *related-to*, *object*⟩ quadruples are returned instead of tuples. If no regular expressions match the current processing sentence, the *object* and *related-to* are null.
We call our new algorithm NECOE which stands for "Named Entity Category Object Extraction".

5. Utilizing quadruples for semantic search

The extracted ⟨*ne*, *category*, *related-to*, *object*⟩ quadruples are valuable for NLP applications. In this section, we use them for answering some types of questions. If *ne* in a quadruple is a person, the quadruple helps answer the query: "Who is *ne*?". If *ne* is of another type, such as *organization* or *location*, the quadruple helps answer the query: "What is *ne*?" For answering the question, we just search for a quadruple having the same *ne* as that of the question. If a quadruple is found, then the answer is:

ne is a *category related-to object*.

The extracted quadruples also help answer *list* questions, e.g., "Give me the list of analyst for Prudential-Bache Securities Inc." The general form of this question type is "Give me the list of *category* [*related-to object*]", where the part in square brackets is optional. For answering this question type, we search for the list *L* of quadruples having the same *category* (*related-to* and *object*) as those of the question. The answer is the list of *nes* of quadruples in *L*.

If *related-to* of a question is 'of', we also search for quadruples whose *object* is embedded in *category* (case (d) as discussed in Section 3.2). For example, if the question is "Give me the list of president of IBM", we also search for quadruples whose *category* is "IBM president".

6. Experiments and evaluation

Since person related questions takes a large portion among NE-related questions, as seen in Text Retrieval Conference (TREC) 9 question-answering track[2], our experiments concentrated on extracting person named-entity-related relations.

6.1. Dataset

We used the same corpus as that used in (Nguyen and Shimazu 2007), i.e., the Wall Street Journal (WSJ) corpus which consists of 595 files. After extracting the body part and removing other parts, e.g., the headers, a plain text collection of nearly 3 million sentences with the size of 308 MB was produced. From this text collection, a test set of 1,000 sentences was randomly selected. From the test set, 385 "*ne* is-a *category*" relations (called is-a relation for short) were manually extracted. Among 385 is-a relations, 199 relations had additional related-to relations. The distribution of related-to relations according to cases discussed in Section 3.2 is given in Table 1. In the preprocessing step of the algorithm in Figure 4, all NEs in this plain text collection were tagged by LingPipe[3]. After removing sentences that contain no person NE, a collection of 667,981 sentences was produced (we call this big-dataset). Next, OpenNLP[4] was used to add POS and chunk tags for the big-dataset.

[2]http://tangra.si.umich.edu/clair/NSIR/cgi-bin/trec-question.cgi?collection=9&script=html/nsir.cgi
[3] http://www.alias-i.com/lingpipe/index.html
[4] http://opennlp.sourceforge.net

Table 1: The distribution of related-to relations.

Case	a)	b)	c)	d)	e)
%	74.87	17.74	2.05	3.59	2.05

Table 2: Results of experiments.

Is-a relations						
Pattern	NECOE-NoValidation			NECOE		
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Seed	89.03	35.84	51.11	99.26	35.06	**51.82**
Exact	63.41	72.47	67.64	94.48	75.58	**83.98**
Sketch	62.88	74.81	68.33	94.50	80.26	**86.80**
Related-to relations						
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Seed	92.13	41.21	**56.94**	97.53	39.70	56.43
Exact	79.34	48.24	60.00	97.03	49.25	**63.97**
Sketch	76.64	52.76	62.50	96.64	57.64	**72.33**

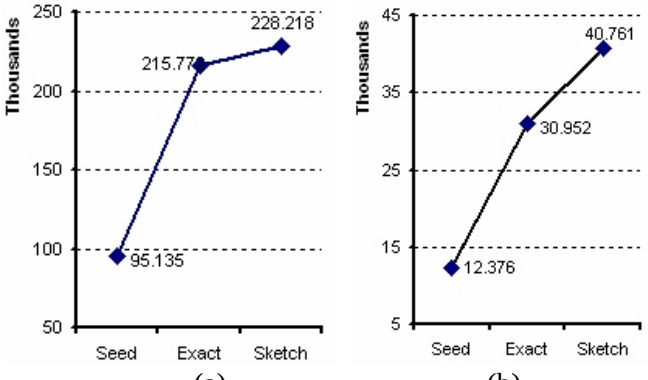
6.2. Experiments and evaluation

Besides NECOE, we also extended the baseline program in (Nguyen and Shimazu 2007) to extract quadruples, and called this program NECOE-NoValidation, since it had no category validation function. We ran NECOE on the big-dataset to get patterns. These patterns are used to extract quadruples on the test set for evaluation. Since a *related-to* relation was extracted after an is-a relation was extracted, we evaluated the results of the two relations in quadruples separately.

PCE used a *threshold* in the pattern generation procedure, and the proper value of threshold selected from experiments was 3. In our experiments, we also set the value of this threshold to 3. The results of our experiments are shown in Table 2. Since extended sketch patterns did not increase the coverage much, we do not show their results.

Figure 6 shows the growth of the number of extracted quadruples and distinct categories from the big-dataset. The figure shows that the number of actual categories (40761) is relatively large.

Though sketch patterns help to extract only 5.5% of the total extracted quadruples, their discovered categories comprise 24% of total distinct categories.



(a) (b)

Figure 6: The growth of extracted quadruples (a) and distinct categories (b).

Table 3: Some top, bottom ranked categories with their frequencies and some related-to relations.

Top ranked	President (22679), Chairman (12835), Analyst (6729), Vice President (6011), Director (5821), Chief Executive Officer (5326), Judge (5050), Dr. (4931), Rep. (3479)
Bottom ranked	part-time CIA employee (1), partnership analyst (1), parliament deputy (1), parts marketing administrator (1), patent specialist (1), personal translator (1), freight carrier (1)
Related-to relation	managing director of investment bank, vice president for economic research, president of Trans World International Inc., deputy of Japanese equities, manager of sales

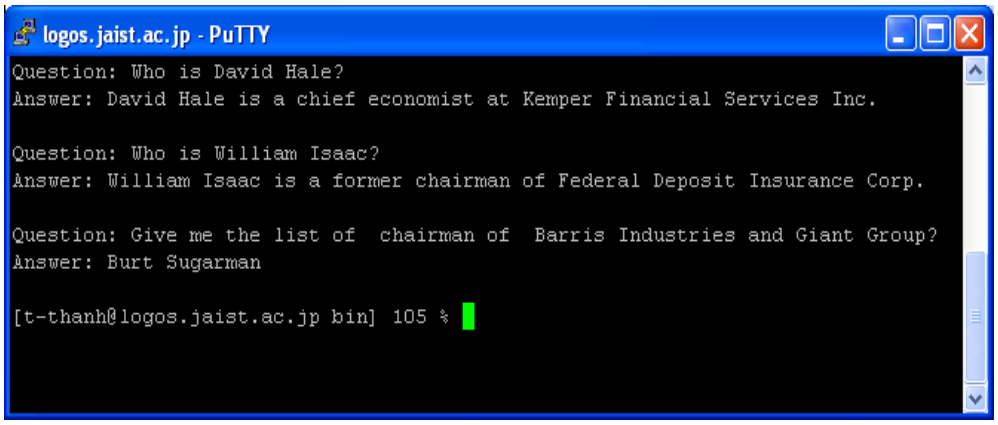


Figure 7: The prototype of a semantic search system.

Table 3 lists some top, bottom ranked categories along with their frequencies, as well as some related-to relations extracted by NECOE.

From our observations, the reason that decreases the precision of related-to relations is derived from the extraction of incorrect is-a relations. The precision of related-to relations is 100%, if it is calculated on correctly extracted is-a relations. Since the related-to relations has a relatively large portion in is-a relations that can not be extracted by PCE, this is the reason that decreases the recall of related-to relations.

Figure 7 introduces the prototype of a semantic search system that utilizes the relations in quadruples extracted from the WSJ corpus.

7. Conclusion

In this paper, we proposed a method for automatically extracting from text documents ⟨*ne*, *category*, *related-to*, *object*⟩ quadruples describing that "*ne* ISA *category*", and "*category* IS-RELATED-TO *object*". We extended PCE algorithm to extract these quadruples. Our experiments on the Wall Street Journal corpus obtained relatively good results.

We also utilize the extracted quadruples in a semantic search system for answering some types of questions.

Our algorithm can be applied to extract quadruples of other NE types, such as *organization* and *location*.

References

Agichtein, E. and L. Gravano, 2000. Snowball: Extracting Relations from Large Plaintext Collections. *Proceedings of the 5th ACM International Conference on Digital Libraries*, pp. 85-94.

Brin, S., 1998. Extracting Patterns and Relations from the World Wide Web. *Proceedings of the 6th International Conference on Extending Database Technology*, pp. 172-183.

Chieu, H. and N. Tou, 2003. Named Entity Recognition with a Maximum Entropy Approach, *Proceedings of Conference on Computational Natural Language Learning 2003* (CoNLL-2003), pp. 160-163.

Fellbaum, C., editor, 1998. WordNet: An Electronic Lexical Database and Some of Its Applications, *MIT Press*.

Nguyen, T. T. and A. Shimazu, 2007. Automatic Extraction of the Fine Category of Person Named Entities from Text Corpora. *IEICE Transactions on Information and Systems, Special section on Knowledge, Information and Creativity Support System*, Vol. E90-D, No. 10, 1542-1549.